

4 Sistema Experimental

O desenvolvimento do sistema experimental requer a integração entre as áreas de mecânica, eletrônica, controle e visão computacional. Através da Figura 17 é possível compreender melhor todas as etapas do projeto do sistema e como ele está estruturado.

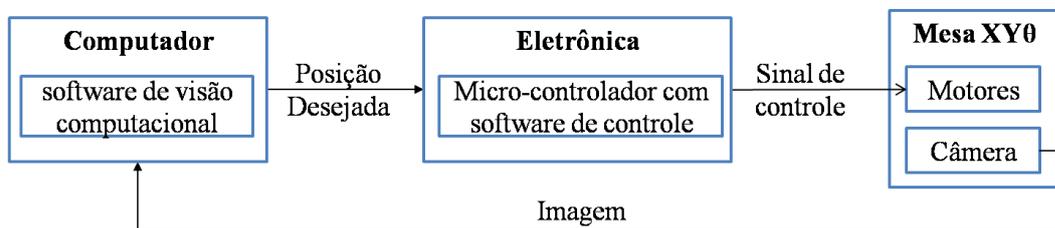


Figura 17 - Esquema do Sistema Experimental

O sistema utiliza uma câmera fixa a uma mesa coordenada XYθ, que irá ser controlada em direção a um objeto alvo. A imagem extraída da câmera é processada por um computador PC, e através de um software de visão computacional são determinadas as posições a serem atingidas pela mesa. Essa posição é enviada a uma placa eletrônica, onde um microcontrolador transforma esta informação em sinais elétricos para os motores/atuadores da mesa.

Este capítulo irá apresentar o procedimento experimental criado para testar e validar os resultados. A primeira seção descreve todo o desenvolvimento mecânico realizado, incluindo a automatização de uma mesa coordenada responsável pela locomoção do protótipo. Em seguida, é apresentado o modelo matemático da mesa, vista como um manipulador robótico. A terceira seção descreve o sistema eletrônico utilizado como interface entre o sistema experimental e o computador. Na quarta seção será apresentado o *software* criado para gerenciar todas as arquiteturas de controle. E por fim os equacionamentos obtidos através do processamento de imagem utilizados dentro do sistema de controle.

4.1. Sistema mecânico

Conforme mencionado no Capítulo 1, o presente trabalho possui potenciais aplicações em tarefas de intervenção submarina. Em uma típica tarefa de interesse da Petrobras, o manipulador Slingsby TA-40 (Figura 18), acoplado a uma ROV, realiza as movimentações necessárias para que a ferramenta em sua extremidade atinja o objeto desejado, no caso válvulas de equipamentos (*manifolds*). Este manipulador hidráulico, de 6 graus de liberdade, é capaz de suportar uma carga de até 210kg.

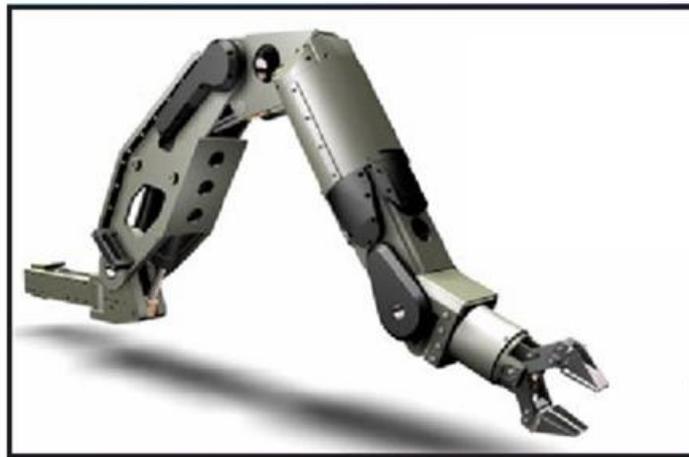


Figura 18 - Manipulador TA-40

A arquitetura fechada do controlador do manipulador TA-40 dificulta a implementação das técnicas de controle propostas sem o suporte do fabricante. Deste modo, optou-se por utilizar uma mesa XY θ , com 3 graus de liberdade, já existente no Laboratório de Robótica da PUC-Rio, para a comprovação experimental. A mesa, apesar de operar no seco, inclui todos os elementos de controle que poderiam ser implementados no manipulador TA-40.

Pela Figura 19, pode-se perceber os 3 motores responsáveis pela movimentação dos 3 elos da mesa. Os dois primeiros elos de translação, são responsáveis pela movimentação em x e y , e o terceiro é referente à rotação. Uma câmera fica acoplada a este último elo, podendo assim ser girada de -180° a 180° .

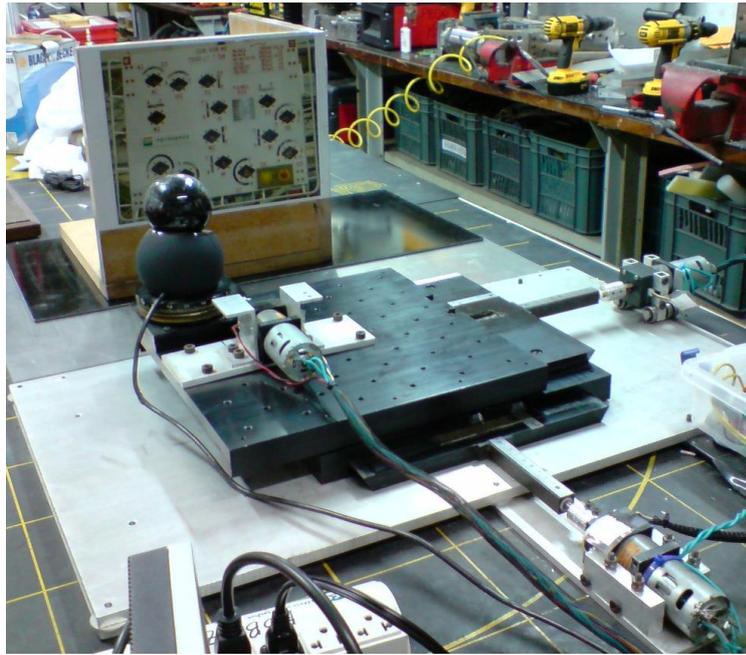


Figura 19 - Mesa Coordenada XY0

A mesa era originalmente controlada manualmente através de três fusos, porém estes não permitiam uma movimentação da mesa de forma automática. Para a automatização, utilizaram-se 3 motoredutores de corrente contínua da marca *Banebots*, todos com tensão máxima de entrada igual a 24 volts, para que, acoplados aos eixos, pudessem ser controlados por uma eletrônica e posteriormente por um computador. As especificações técnicas e o *datasheet* dos motores encontram-se no Apêndice I.



Figura 20 – Motoredutor *Banebots*

Para as juntas referentes às coordenadas x e y , utilizaram-se motores com redução 16:1. Já para o terceiro elo, referente à angulação da câmera, optou-se pelo motor com redução de 125:1, uma vez que não se fazia necessária alta velocidade de rotação. Foram projetados acoplamentos mecânicos em forma

cilíndrica, conectando os eixos do motor e do fuso da mesa, os quais possuem diferentes diâmetros.

Para manter os motores fixos à mesa, foram realizadas algumas modificações na estrutura do sistema, de forma que não comprometesse a movimentação dos elos. É necessário destacar que o motor referente à movimentação na coordenada x não pode ser fixado à mesa. Isto acontece pois quando o motor referente à movimentação em y é acionado, o posicionamento do motor x muda (conforme a Figura 22). Logo, para que o motor x possa deslizar sobre a mesa, foram acoplados roletes a ele. Na Figura 21, pode-se perceber o acoplamento mecânico construído para unir os elos do motor e da mesa, assim como os roletes presos na base do motor, permitindo assim o seu deslocamento.

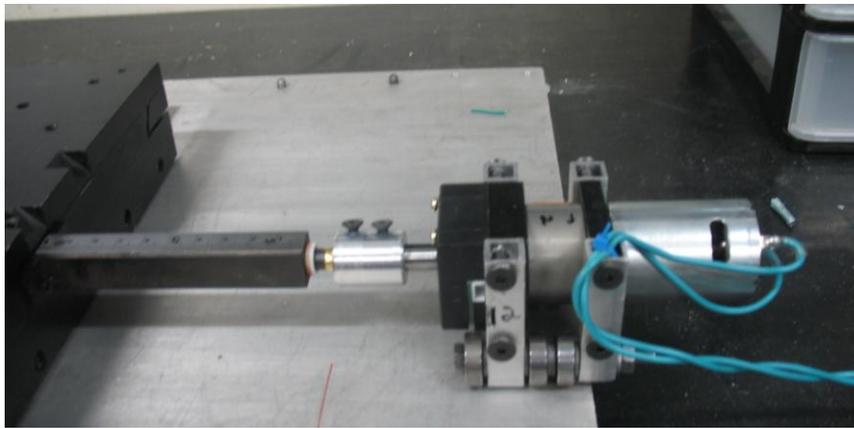


Figura 21 - Acoplamento mecânico entre os eixos do motor e da mesa

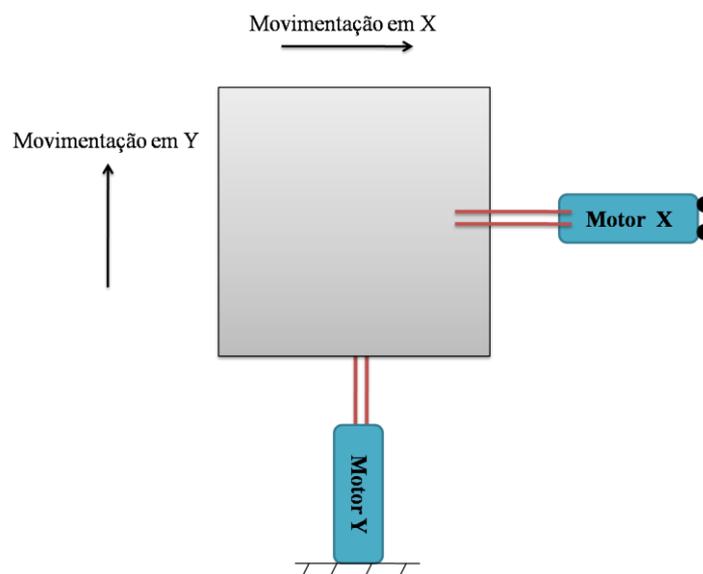


Figura 22 - Esquema da Mesa XY θ

Na extremidade da mesa, foi posicionada uma câmera digital para fazer a aquisição da imagem. Foi escolhida a câmera QuickCam® Orbit AF da marca Logitech, e esta foi acoplada a mesa juntamente com seu suporte próprio (Figura 23). Esta câmera foi escolhida para este projeto pela alta qualidade das imagens extraídas, podendo capturar vídeos de até 960x720 *pixels*, em um total de 30 frames/s. Além da alta qualidade, a câmera possui foco automático, altamente necessário devido à variação de sua posição em relação ao objeto. Por ser uma câmera do tipo *webcam*, sua conexão ao computador é feita através de porta USB, facilitando assim a comunicação.

A câmera possibilita extrair imagens em diversos padrões, porém neste trabalho optou-se por utilizar imagens coloridas, no formato RGB. O tamanho das imagens extraídas variou entre 352x288 e 960x720 *pixels* dependendo do controle utilizado.



Figura 23 - Câmera Logitech utilizada para aquisição das imagens

4.2. Modelo Matemático da Mesa XY θ

Para estabelecer o modelo matemático da mesa XY θ , ela é representada como um manipulador plano de 3 graus de liberdade. As movimentações em x e y podem ser comparadas a juntas prismáticas, sendo suas coordenadas representadas por distâncias d_1 e d_2 respectivamente (vide Figura 24). Já a rotação da câmera na extremidade pode ser interpretada como uma junta rotativa com ângulo representado por θ_3 em relação ao eixo x . São definidos d_1 , d_2 e θ_3 como positivos no sentido em que crescem de valor em relação a x , y e no

sentido trigonométrico. A Figura 24 apresenta o Esquema do manipulador juntamente com os parâmetros de cada elo.

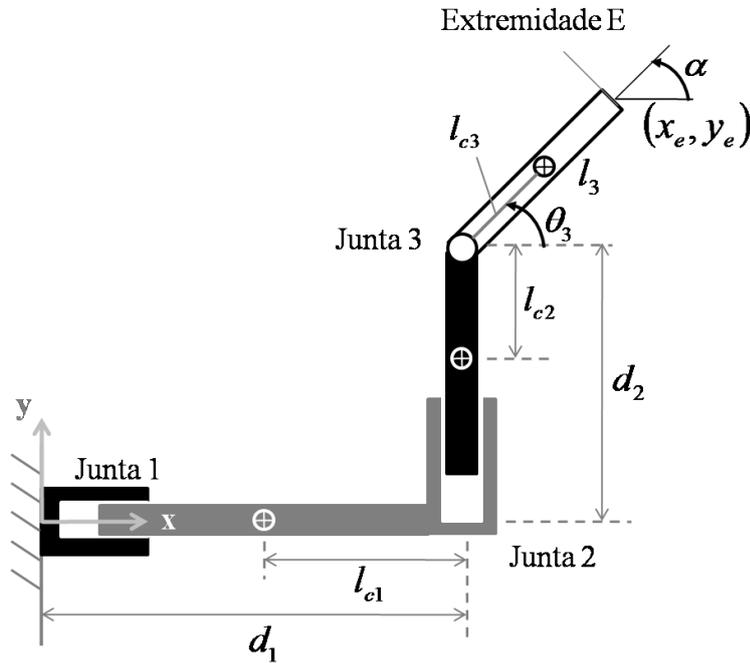


Figura 24 - Parâmetros do Manipulador

Assume-se que os elos 1, 2 e 3 possuem respectivamente massa m_1 , m_2 e m_3 , momento de inércia em relação ao centro de massa I_1 , I_2 e I_3 , e a distância entre seu centro de massa e as juntas 2, 3 e 3 são l_{c1} , l_{c2} e l_{c3} . Na mesa coordenada, como o centro de massa da câmera está sobre o centro de rotação da junta 3, $l_{c3} = 0$.

A extremidade do manipulador é definida pelas coordenadas (x_e, y_e, α) e, a partir da Figura 24, pode-se dizer que

$$x_e = d_1 + l_3 \cos(\theta_3) \quad (39)$$

$$y_e = d_2 + l_3 \sin(\theta_3) \quad (40)$$

$$\alpha = \theta_3 \quad (41)$$

A equação de movimento de cada elo é dada pelas seguintes equações:

$$\tau_i = \sum_j H_{ij} \ddot{q}_j + \sum_j \sum_k h_{ijk} \dot{q}_j \dot{q}_k + G_i, \quad i=1, 2 \quad (42)$$

sendo H a matriz de inércia do manipulador, h termos não-lineares responsáveis por efeitos como a força centrífuga e termo de Coriolis, G refere-se ao torque que

deve ser adicionado ao elo para compensar os efeitos gravitacionais, e q

$$\text{representa o vetor de posição dos elos } q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \theta_3 \end{bmatrix}.$$

Para descobrir as equações do movimento dos elos, é necessário primeiro, calcular a matriz de inércia do manipulador. Como o sistema é plano e só possui uma junta rotatória, os jacobianos analítico e geométrico são iguais. Inicia-se então o cálculo das matrizes Jacobianas que relacionam a velocidade da extremidade com a velocidade das juntas do robô. Estas podem ser divididas em uma parte linear J_L :

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & -l_3 \sin(\theta_3) \\ 0 & 1 & l_3 \cos(\theta_3) \end{bmatrix}}_{J_L} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (43)$$

e angular J_A :

$$\begin{bmatrix} \dot{\alpha} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}}_{J_A} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (44)$$

É necessário também calcular todas as matrizes Jacobianas totais $J_L^{(i)}$ e $J_A^{(i)}$ (Asada e Slotine, 1985) que relacionam a velocidade linear e angular do centro de massa de cada elo às velocidades das juntas.

Para a junta 1 tem-se que:

$$\begin{pmatrix} x_{c1} \\ y_{c1} \end{pmatrix} = \begin{pmatrix} d_1 - l_{c1} \\ 0 \end{pmatrix} \rightarrow J_L^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (45)$$

$$\alpha_1 = 0^\circ \rightarrow J_A^{(1)} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \quad (46)$$

para a junta 2:

$$\begin{pmatrix} x_{c2} \\ y_{c2} \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 - l_{c2} \end{pmatrix} \rightarrow J_L^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (47)$$

$$\alpha_2 = 0^\circ \rightarrow J_A^{(2)} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \quad (48)$$

e por último os termos relacionados à junta 3:

$$\begin{pmatrix} x_{c3} \\ y_{c3} \end{pmatrix} = \begin{pmatrix} d_1 + l_{c3} \cos(\theta_3) \\ d_2 + l_{c3} \sin(\theta_3) \end{pmatrix} \rightarrow J_L^{(3)} = \begin{bmatrix} 1 & 0 & -l_{c3} \sin(\theta_3) \\ 0 & 1 & l_{c3} \cos(\theta_3) \end{bmatrix} \quad (49)$$

$$\alpha_3 = \theta_3 \rightarrow J_A^{(3)} = [0 \quad 0 \quad 1] \quad (50)$$

Pode-se então determinar a matriz de inércia do manipulador, expressa de maneira geral como $H = \sum_{i=1}^n (J_L^{iT} \cdot J_L^i \cdot m_i + J_A^{iT} \cdot I_i \cdot J_A^i)$. Para o caso do manipulador, tem-se que

$$H = \begin{bmatrix} m_1 + m_2 + m_3 & 0 & -m_3 l_{c3} \sin(\theta_3) \\ 0 & m_2 + m_3 & m_3 l_{c3} \cos(\theta_3) \\ -m_3 l_{c3} \sin(\theta_3) & m_3 l_{c3} \cos(\theta_3) & m_3 l_{c3}^2 + I_3 \end{bmatrix} \quad (51)$$

Para o cálculo dos termos h , também presentes nas equações de movimento, sabe-se de maneira geral que

$$h_{ijk} = \frac{\partial H_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial H_{jk}}{\partial q_i} \quad (52)$$

Sendo assim, para o caso da mesa XY θ , os seis termos são expressados como:

$$\begin{aligned} h_{133} &= -m_3 l_{c3} \cos(\theta_3) & h_{233} &= -m_3 l_{c3} \sin(\theta_3) \\ h_{313} &= -\frac{m_3 l_{c3} \cos(\theta_3)}{2} & h_{323} &= -\frac{m_3 l_{c3} \sin(\theta_3)}{2} \\ h_{331} &= \frac{m_3 l_{c3} \cos(\theta_3)}{2} & h_{332} &= \frac{m_3 l_{c3} \sin(\theta_3)}{2} \end{aligned} \quad (53)$$

Sabendo que neste caso os elos encontram-se paralelos a mesa, supõem-se os efeitos gravitacionais (representados pelos termos G_i) desprezíveis. Sendo assim, todos os termos necessários para o cálculo das equações de movimento já foram determinados e, a partir da equação (46) são obtidas as equações de movimento:

$$F_1 = (m_1 + m_2 + m_3) \ddot{d}_1 - m_3 l_{c3} \sin(\theta_3) \ddot{\theta}_3 - m_3 l_{c3} \cos(\theta_3) \dot{\theta}_3^2 \quad (54)$$

$$F_2 = (m_2 + m_3) \ddot{d}_2 + m_3 l_{c3} \cos(\theta_3) \ddot{\theta}_3 - m_3 l_{c3} \sin(\theta_3) \dot{\theta}_3^2 \quad (55)$$

$$\tau_3 = -m_3 l_{c3} \sin(\theta_3) \ddot{d}_1 + m_3 l_{c3} \cos(\theta_3) \ddot{d}_2 + (I_3 + m_3 l_{c3}^2) \ddot{\theta}_3 \quad (56)$$

Uma vez determinadas as equações da dinâmica dos três elos, é possível simular a trajetória do manipulador. Neste trabalho, no entanto, não foram realizadas simulações, pois as equações referentes à parte de visão computacional não poderiam ser simuladas. Isso porque a simulação dependeria das imagens obtidas durante a execução do controle em tempo real.

Para aplicar acelerações constantes a_x (linear) na junta 1, a_y (linear) na junta 2, e α' (angular) na junta 3 é necessário utilizar um controlador não linear para cancelar os efeitos dinâmicos. Deseja-se controlar uma trajetória com valores desejados $d_{1d}(t) = a_x t^2 / 2$, $d_{2d}(t) = a_y t^2 / 2$ e $\theta_{3d}(t) = \alpha' t^2 / 2$ supondo sem perda de generalidade, a mesa inicialmente parada na posição $(d_1, d_2, \theta_3) = (0, 0, 0)$ no instante $t = 0$. Pode-se então determinar a lei não-linear de controle de torque computado combinado com um controle PID de ganhos K_{Pi} , K_{Ii} e K_{Di} . Sabendo que a lei de controle PID, obedece à formulação

$$u_i = K_{Pi}(d_{id} - d_i) + K_{Di}(\dot{d}_{id} - \dot{d}_i) + K_{Ii} \int (d_{id} - d_i) dt \quad (57)$$

tem-se para os três elos

$$u_1 = K_{P1} \left(\frac{a_x t^2}{2} - d_1 \right) + K_{D1} (a_x t - \dot{d}_1) + K_{I1} \int (a_x - \ddot{d}_1) dt \quad (58)$$

$$u_2 = K_{P2} \left(\frac{a_y t^2}{2} - d_2 \right) + K_{D2} (a_y t - \dot{d}_2) + K_{I2} \int (a_y - \ddot{d}_2) dt \quad (59)$$

$$u_3 = K_{P3} \left(\frac{\alpha' t^2}{2} - \theta_3 \right) + K_{D3} (\alpha' t - \dot{\theta}_3) + K_{I3} \int (\alpha' - \ddot{\theta}_3) dt \quad (60)$$

Obtidas as leis de controle, pode-se então calcular o torque a ser enviado a cada junta:

$$F_1 = (m_1 + m_2 + m_3)u_1 - m_3 l_{c3} \sin(\theta_3)u_3 - m_3 l_{c3} \cos(\theta_3)\dot{\theta}_3^2 \quad (61)$$

$$F_2 = (m_2 + m_3)u_2 + m_3 l_{c3} \cos(\theta_3)u_3 - m_3 l_{c3} \sin(\theta_3)\dot{\theta}_3^2 \quad (62)$$

$$\tau_3 = -m_3 l_{c3} \sin(\theta_3)u_1 + m_3 l_{c3} \cos(\theta_3)u_2 + (I_3 + m_3 l_{c3}^2)u_3 \quad (63)$$

O modelo apresentado acima representa o controle dinâmico do manipulador, conhecido também como controle de Torque Computado. Através deste são considerados efeitos não-lineares da dinâmica do robô originários da velocidade dos elos ou da influência entre eles. No entanto estes efeitos não são significativos na maioria dos sistemas *eye-in-hand*, cujas velocidades são limitadas pelo tempo de processamento das imagens. Sendo assim, para este trabalho foi apenas considerado o controle PID, utilizando apenas o modelo cinemático do manipulador.

4.3. Sistema eletrônico

O sistema eletrônico desenvolvido para este projeto encarrega-se de receber uma posição desejada do computador, processá-la, e enviar o torque necessário para os motores.

Duas maneiras de se fazer um controle eletrônico de atuadores pelo computador foram consideradas. Na primeira, uma porta paralela é utilizada para controlar conversores Analógicos/Digitais que fornecem uma tensão analógica de referência para que um circuito de alta potência faça a amplificação de corrente e/ou tensão.

Na segunda, uma porta de comunicação serial (USB, RS232, etc.) é utilizada para enviar valores que serão interpretados por uma eletrônica microcontrolada (provida de uma interface de comunicação serial) que comandará circuitos de alta potência para ativação dos atuadores. Neste caso, o microcontrolador utilizado pode inclusive ser programado para receber do computador somente as posições desejadas das juntas, executando o controle PID internamente.

Desenvolveu-se para este trabalho uma placa que se comunica com o computador via serial, contendo esta um microcontrolador (Apêndice II) responsável por calcular e enviar torques para os motores. A placa eletrônica desenvolvida funciona como uma interface entre o computador e os motores de ativação da mesa. A Figura 25 apresenta um Esquema da placa, e através dela nota-se a presença do conector serial (identificado pelo número 1) e do microcontrolador (identificado pelo número 2). Foram incluídos também botões de ajuste manual de posição. Cada par de botões é responsável pelo acionamento de um motor no sentido positivo e negativo. Os seis botões (dois para cada um dos três elos) também estão identificados na Figura 25 pelo número 3.

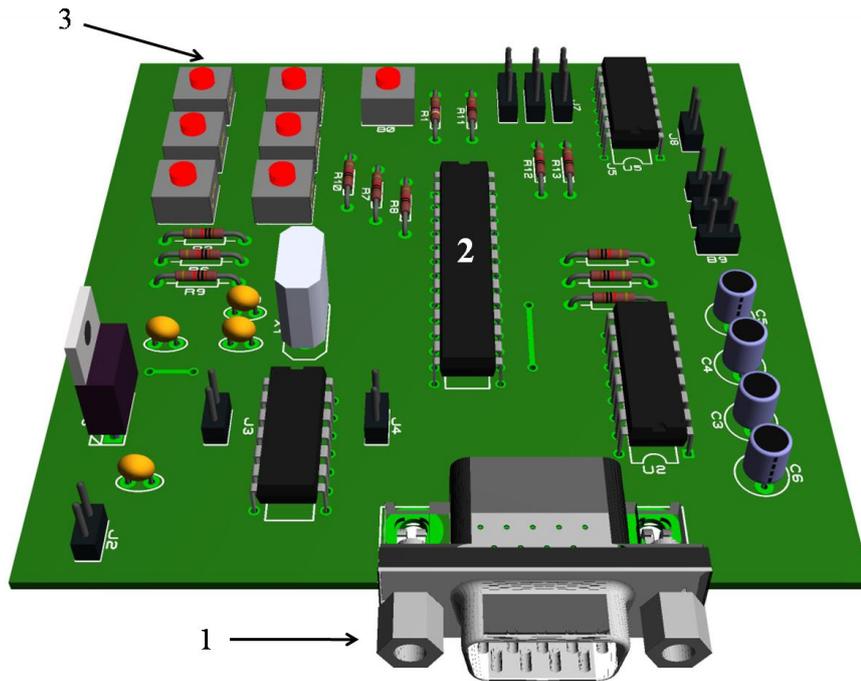


Figura 25 - Placa eletrônica

O sinal de saída da eletrônica é do tipo PPM (*Pulse Position Modulation*). Este sinal foi escolhido porque diversos sistemas de potência disponíveis no mercado utilizam-no como sinal de ativação. Desta forma, a mesma eletrônica pode ser utilizada em diferentes sistemas de potência, se adequando assim a corrente drenada por diferentes motores DC.

Para ativação dos motores DC, foram utilizados controladores de velocidade Banebots® BB-12-45, que suportam correntes contínuas de 12A, 45A de pico, e são ativados por sinais PPM vindos diretamente da placa eletrônica.



Figura 26 - Controlador de Velocidade *Banebots*

Neste projeto serão estudadas duas diferentes técnicas de controle (conforme visto no Capítulo 3). O controle *look-and-move* se utiliza de sensores de posição para realimentar o sistema. Já o controle servo-visual desconhece a posição da mesa no espaço e utiliza apenas o erro entre o objeto e a câmera (obtido através de imagens) para levar a mesa coordenada até a posição desejada. Para que a eletrônica desenvolvida pudesse se adequar às duas técnicas, foram implementadas pequenas mudanças no microcontrolador responsável pelo controle. A implementação no sistema eletrônico de ambas as técnicas serão descritas nas seções seguintes.

4.3.1. Sistema com realimentação por sensores de posição (Controle *Look-and-Move*)

Uma vez recebida do computador a informação da posição desejada, calculada a partir da imagem da câmera na posição inicial, o microcontrolador lê a posição atual dos motores através dos sensores de posição das juntas, e calcula através do controle PID o torque a ser enviado para cada motor. Este ciclo se repete a cada instante de tempo, até que a mesa atinja a posição desejada. Este processo é inteiramente realizado dentro do microcontrolador, sem que seja necessária qualquer outra intervenção do *software* do computador, e sem a captura de novas imagens. Se, durante a execução da tarefa, alguma outra informação for enviada à eletrônica, o controle automaticamente muda de trajetória, dirigindo-se então à nova posição desejada. Vale ressaltar que as constantes K_p , K_I e K_D , necessárias para o controle PID, são informadas continuamente à placa eletrônica juntamente com a informação da posição desejada, permitindo assim que os ganhos sejam modificados se necessário.

4.3.2. Sistema com realimentação por imagem (Controle Servo-Visual)

No controle servo-visual, o erro entre a posição desejada e a real não é obtido no microcontrolador, e sim no computador. Isso acontece pois o erro entre a posição desejada e a real é obtido através das imagens, e varia ao longo da

movimentação da mesa. Para este caso, o cálculo do controle PID é feito através do *software* do computador, e apenas o resultado é enviado à placa eletrônica.

Para comportar os dois sistemas de controle, a formulação do PID utilizada no microcontrolador foi acrescentada de um novo fator:

$$u_i = K_{P_i}(d_{id} - d_i) + K_{D_i}(\dot{d}_{id} - \dot{d}_i) + K_{I_i} \int (d_{id} - d_i) dt + T_i \quad (64)$$

onde T_i é um torque calculado pelo *software* e enviado à eletrônica. Para que este torque seja enviado diretamente aos motores sem ser alterado pela rotina PID residente no micro-controlador, é necessário que os valores das posições desejadas e as constantes K_P , K_I e K_D enviados ao sistema eletrônico sejam nulos, resultando em $u_i = T_i$.

No caso do controle *look-and-move*, os valores das posições desejadas e as constantes do controle PID são enviados normalmente, com a diferença que para este caso os valores de T_i devem ser iguais a zero. Na Tabela 1 estão listados, em ordem de envio, os parâmetros enviados à eletrônica, variando com as técnicas de controle.

Tabela 1 - Parâmetros enviados através da porta serial à eletrônica desenvolvida

Parâmetros enviados à eletrônica		Controle <i>look-and-move</i>	Controle Servo-visual
Posição desejada	Elo1	$x_d \neq 0$	$x_d = 0$
	Elo2	$y_d \neq 0$	$y_d = 0$
	Elo3	$\theta_d \neq 0$	$\theta_d = 0$
Ganhos (iguais para os três elos)	Proporcional	$K_P \neq 0$	$K_P = 0$
	Integral	$K_I \neq 0$	$K_I = 0$
	Derivativo	$K_D \neq 0$	$K_D = 0$
Torque enviado aos motores	Elo1	$T_1 = 0$	$T_1 \neq 0$
	Elo 2	$T_2 = 0$	$T_2 \neq 0$
	Elo 3	$T_3 = 0$	$T_3 \neq 0$

Vale ressaltar que os ganhos utilizados no controle servo-visual não estão representados na tabela acima. Seus valores diferentes de zero, são calibrados dentro do software, para gerar os torques T_1 , T_2 e T_3 .

Tanto para o controle *look-and-move* quanto o servo-visual, o controle PID foi empregado em sua forma discretizada. O taxa de amostragem varia com a frequência ora do microcontrolador (controle *look-and-move*), ora do computador (controle servo-visual). A formulação geral do controle PID discreto é

$$u(t) = K_p e(t) + \frac{K_D}{\Delta t} (e(t) - e(t-1)) + K_I \Delta t \sum e(t) \quad (65)$$

onde o erro $e(t)$ representa a diferença entre a posição desejada e a atual. Conforme a equação (65), o termo derivativo pode ser representado como a diferença entre o erro no instante atual e o instante anterior. Já o termo integral representa um acumulador, em que a cada instante de tempo é acrescido do valor do erro.

4.4. Software de controle

Para a implementação do *software* de controle no computador optou-se por utilizar o programa Matlab® da empresa *MathWorks*, devido à sua vasta biblioteca de visão computacional, e também pela facilidade de comunicação com a porta serial. O *software* criado poderá ser de grande ajuda na área de pesquisa, servindo como ferramenta de estudo.

A tela principal do programa, apresentada na Figura 27, contém todas as opções de controle a serem executadas. No canto direito da tela, uma chave indica se o alvo utilizado será um círculo vermelho (utilizado nos primeiros testes), ou se será utilizada uma imagem de um painel de válvulas típico de intervenções submarinas (simulando um painel real).

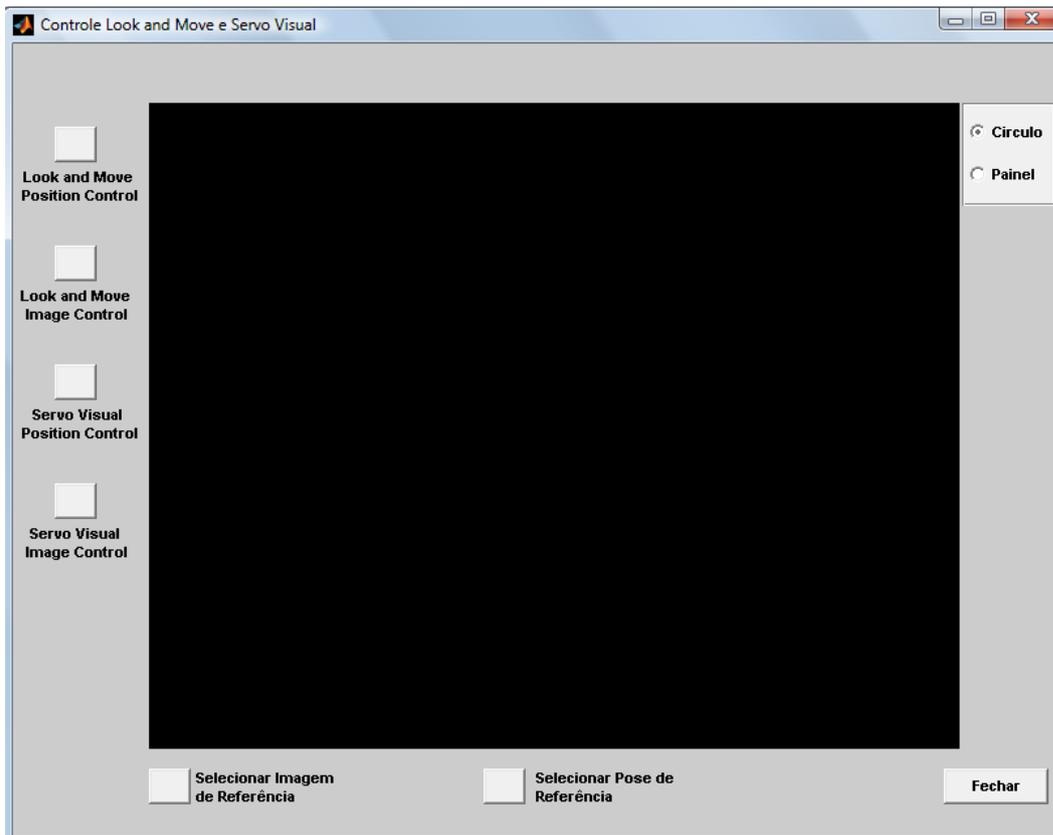


Figura 27 - Tela principal do *software*

Na parte inferior da tela, dois botões permitem ao usuário a troca das variáveis de referência. No caso do controle baseado em imagem, uma janela permite a escolha da imagem de referência (Figura 28). Ou seja, a imagem da câmera que se deseja observar quando o sistema atingir a posição final. No caso do controle baseado em pose, uma janela permite ao usuário que indique as posições desejadas em x , y e θ , representadas relativamente ao objeto de interesse (Figura 29).

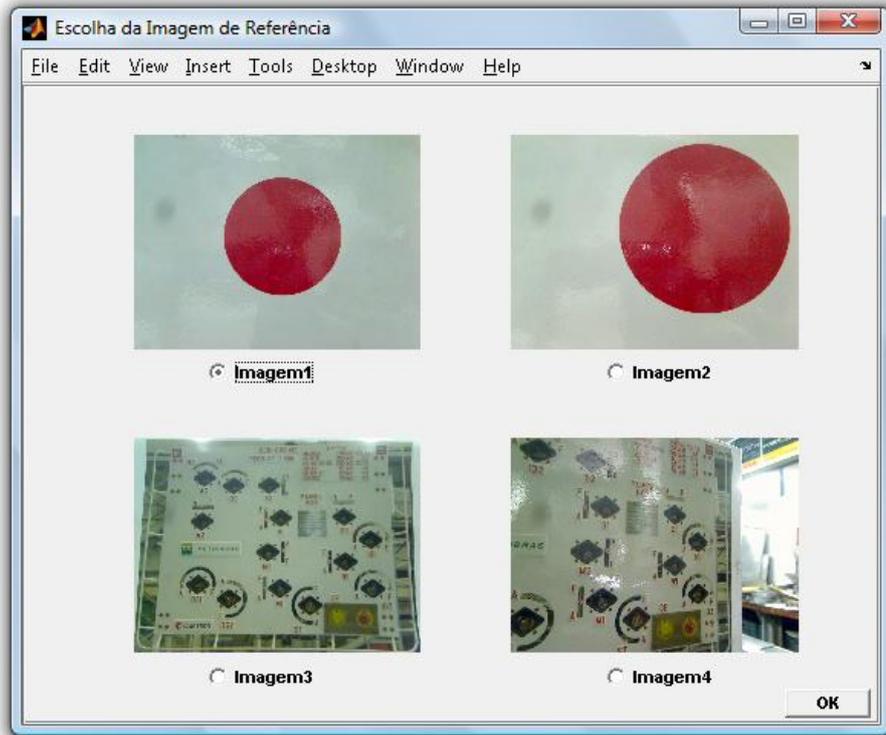


Figura 28 - Janela de seleção da imagem de referência (controle baseado em imagem)

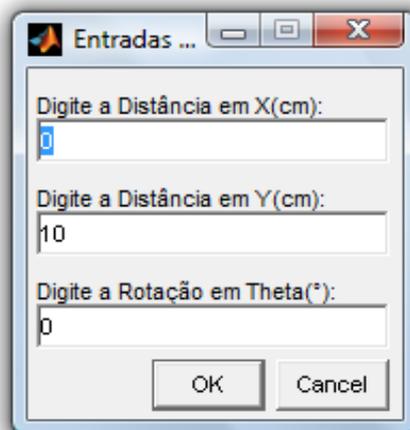


Figura 29 - Janela de distâncias desejadas do objeto de interesse (controle baseado em pose)

Uma vez escolhida a imagem ou pose de referência, e se o controle irá se aplicar ao círculo ou ao painel, o usuário deverá escolher na lateral esquerda da tela entre as opções de controle:

- Controle *look-and-move* baseado em pose

- Controle *look-and-move* baseado em imagem
- Controle servo-visual baseado em pose
- Controle servo-visual baseado em imagem

A seguir será apresentado um passo a passo do programa, descrevendo todas as principais etapas efetuadas pelo *software* de controle.

- Passo1: Captura uma imagem da câmera.
- Passo2: Controle baseado em pose ou em imagem.
 - Baseado em pose:
 - Processa imagem obtida no passo1, e obtém coordenadas relativas entre a câmera e o objeto: $x_{rel}, y_{rel}, \theta_{rel}$.
 - Usuário determina posição de referência desejada $x_{ref}, y_{ref}, \theta_{ref}$.
 - Se controle escolhido for **look-and-move**:
 - ✓ Lê posição atual dos *encoders* x_0, y_0, θ_0 .
 - ✓ Calcula posição desejada: $x_d = x_0 + (x_{rel} - x_{ref})$,
 $y_d = y_0 + (y_{rel} - y_{ref})$, $\theta_d = \theta_0 + (\theta_{rel} - \theta_{ref})$.
 - ✓ Envia parâmetros para eletrônica.
 - ✓ Fim.
 - Se controle escolhido for **servo-visual**:
 - ✓ Obtém-se o erro entre a câmera e o objeto:
 $e_x = x_{rel} - x_{ref}$, $e_y = y_{rel} - y_{ref}$, $e_\theta = \theta_{rel} - \theta_{ref}$.
 - ✓ Cálculo do controle PID, obtendo torques para as 3 juntas.
 - ✓ Envia parâmetros para eletrônica.
 - ✓ Se erro em alguma das 3 juntas for maior que um limiar, retorna ao passo1 para capturar uma nova imagem, senão, fim.
 - Baseado em imagem
 - Determinam-se as características da imagem obtida no
Passo1: S_1, S_2, S_3 .

- Compara características obtidas com as características da imagem de referência $\$_{1ref}, \$_{2ref}, \$_{3ref}$.
- A partir da diferença entre as características desejadas e as reais, utiliza-se a matriz Jacobiana de imagem, definida na seção 4.6, para obter as coordenadas relativas entre a

$$\begin{pmatrix} x_{rel} \\ y_{rel} \\ \theta_{rel} \end{pmatrix} = J^* \cdot \begin{pmatrix} \$_1 - \$_{1ref} \\ \$_2 - \$_{2ref} \\ \$_3 - \$_{3ref} \end{pmatrix}.$$

- Se controle escolhido for **look-and-move**:
 - ✓ Lê posição atual dos *encoders* x_0, y_0, θ_0 .
 - ✓ Calcula posição desejada: $x_d = x_0 + x_{rel}$,
 $y_d = y_0 + y_{rel}$, $\theta_d = \theta_0 + \theta_{rel}$.
 - ✓ Envia parâmetros para eletrônica.
 - ✓ Fim.
- Se controle escolhido for **servo-visual**:
 - ✓ Obtém-se erro entre a câmera e o objeto $e_x = x_{rel}$,
 $e_y = y_{rel}$, $e_\theta = \theta_{rel}$.
 - ✓ Inicia cálculo do controle PID, e obtém torques para as 3 juntas.
 - ✓ Envia parâmetros para eletrônica.
 - ✓ Se erro em alguma das 3 características $\$_1, \$_2$ ou $\$_3$ em relação a $\$_{1ref}, \$_{2ref}$ ou $\$_{3ref}$ for maior que um limiar, retorna ao passo1, senão, fim.

Vale ressaltar que durante o controle *look-and-move* a posição inicial (x_0, y_0, θ_0) medida pelos sensores de posição das juntas não precisa ser utilizada no *software* do computador, pois ela é computada diretamente na placa eletrônica.

4.5. Processamento da Imagem

Durante os testes realizados neste trabalho, foram utilizados dois objetos alvos distintos. O primeiro objeto utilizado foi um círculo (vermelho) impresso sobre uma superfície vertical, a fim de facilitar o processamento da visão computacional. Em seguida, foram implementadas novas técnicas que permitiram ampliar o programa, de forma que este funcionasse também utilizando qualquer outro objeto plano como referência. Para validar os experimentos com um alvo genérico, utilizou-se uma imagem de um painel de válvulas submarino, simulando assim uma aplicação real para o projeto. A seguir serão descritos os processamentos das imagens para os dois casos. Primeiro serão apresentados os equacionamentos utilizados no caso do círculo e em seguida para o caso genérico de um objeto plano.

4.6. Equacionamento do sistema com alvo circular

A fim de minimizar o esforço computacional nesta etapa do projeto, foi escolhido um disco circular vermelho, liso, em um plano vertical, e que pudesse ser de fácil reconhecimento na imagem. A cor vermelha foi escolhida aleatoriamente por se tratar de uma das cores primárias, uma vez utilizando o padrão RGB. A partir então da relação entre semi-eixos da elipse resultante na imagem, as distâncias em x , y , θ podem ser conhecidas. Na Figura 30 pode-se observar os parâmetros relevantes neste experimento.

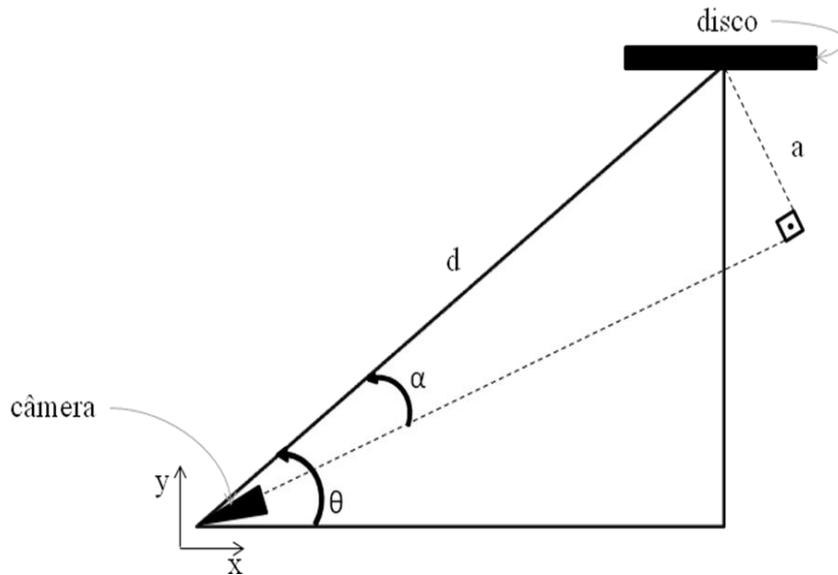


Figura 30 - Esquema dos testes com disco

A Figura 30 mostra a vista superior do disco e da mesa, e admite-se o disco com eixo axial na direção y ; x e y representam os eixos de movimentação da mesa; d é a distância entre o centro da câmera e o centro do disco; a é a distância entre o centro do disco e o eixo óptico da câmera; θ é o ângulo entre o eixo x e o segmento que contém d ; e α é o ângulo entre o eixo óptico e o segmento que contém d . Note que quando o disco encontra-se centralizado na imagem, $a = 0$.

Pela Figura 30, podem ser deduzidas as seguintes equações:

$$x = d \cos \theta \quad (66)$$

$$y = d \sin \theta \quad (67)$$

$$\sin \alpha = \frac{a}{d} \quad (68)$$

Para simplificar as equações, define-se também:

$$\theta' = \theta - \alpha \quad (69)$$

Quando o eixo óptico da câmera atravessa o centro do disco, pode-se perceber uma distância i_a entre o centro do disco e o centro da imagem. Pela Figura 31 é possível perceber também que quando a câmera não se encontra perpendicular ao disco, este pode ser representado na imagem por uma elipse com semi-eixos menor i_r e maior i_R .

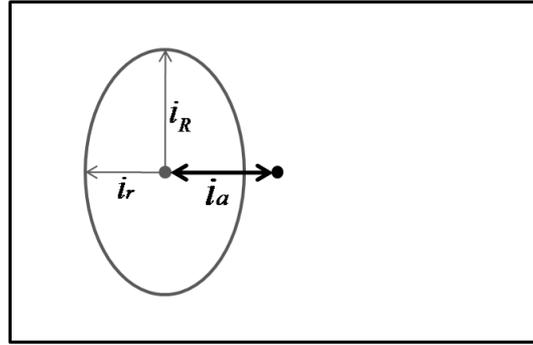


Figura 31 - Imagem obtida da câmera com disco deslocado do centro

Assumindo que o centro da câmera estará sempre na mesma altura que o centro do disco, uma vez que a mesa não se desloca na vertical, pode-se afirmar que o semi-eixo maior só irá variar de tamanho na imagem quando houver deslocamento em y . Ou seja, quando a câmera se aproximar do disco i_r aumenta, e quando a câmera se afastar este diminui. Essa afirmativa só é válida, no entanto, se $a \ll d$, resultando em

$$d = \frac{K}{i_r} \quad (70)$$

sendo K uma constante. Caso a não seja muito menor que d , a expressão acima se torna uma aproximação, adotada aqui por simplicidade, pois nos experimentos conduzidos o valor desejado de a é zero, satisfazendo $a = 0 \ll d$.

Admitindo-se o disco descentralizado da imagem, estabelece-se que:

$$\frac{a}{r} = \frac{i_a}{i_r} \quad (71)$$

onde r é o raio (real) do disco. Para uma imagem centralizada, $i_a = 0$, logo $a = 0$.

Para determinar o ângulo de inclinação do disco em relação à câmera, utiliza-se a razão entre seus semi-eixos na imagem. Observando a Figura 32, onde a imagem da esquerda representa o disco visto de frente, e a imagem da direita a visão superior, pode-se estabelecer que

$$i_r = i_R \sin \theta \Rightarrow \theta = \sin^{-1} \left(\frac{i_r}{i_R} \right) \quad (72)$$

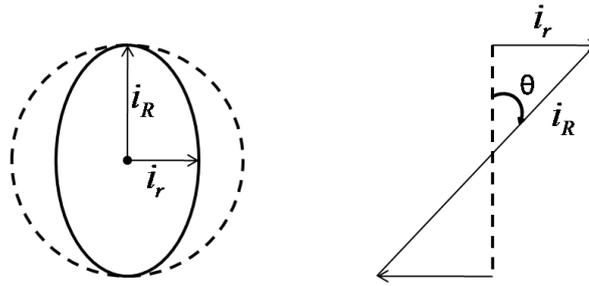


Figura 32 - Vista frontal e superior do disco

Substituindo as equações (68) e (72) em (69), é possível determinar o valor de θ' , obtendo assim os valores necessários para a atuação dos 3 elos x , y e θ , onde $\theta = \theta' + \alpha$. Estes valores, que representam os deslocamentos e a rotação da câmera em relação ao disco, são suficientes para permitir que um controle baseado em pose seja executado.

Para as técnicas de controle baseadas em imagem, as variáveis desejadas são características a serem definidas pelas variáveis $\$1$, $\$2$ e $\$3$. Tais características são escritas, segundo quaisquer características providas da imagem. Para este objeto alvo circular, as variáveis serão escolhidas em função de i_a , i_r e i_R .

A fim de facilitar os cálculos, as variáveis $\$1$, $\$2$ e $\$3$ serão definidas como

$$\$1 = \frac{1}{i_R}, \quad \$2 = \frac{i_r}{i_R}, \quad \$3 = \frac{i_a}{i_R}.$$

Reescrevendo as equações já obtidas em função das novas variáveis, tem-se que a equação (70) passa a ser escrita como $d = \frac{K}{i_R} = K \$1$ e a equação (72) por

$$\sin(\theta) = \left(\frac{i_r}{i_R} \right) = \$2. \text{ Sendo assim, os valores de } x \text{ e } y, \text{ descritos anteriormente}$$

nas equações (66) e (67), passam a ser escritos como:

$$x = K \$1 \sqrt{1 - \$2^2} \quad (73)$$

$$y = K \$1 \$2 \quad (74)$$

A equação (71), $\frac{a}{r} = \frac{i_a}{i_R}$, pode ser reescrita como $\$3 = \frac{a}{r}$; a equação (68),

$\text{sen } \alpha = \frac{a}{d}$, como $\text{sen } \alpha = \frac{r \$3}{K \$1}$; e, por último, a equação (69), $\theta' = \theta - \alpha$, como

$$\theta' = \text{sen}^{-1}(\$2) - \text{sen}^{-1}\left(\frac{r \$3}{K \$1}\right).$$

Sendo J_{sp} a matriz jacobiana de imagem responsável pela transformação entre o vetor de características $\$ = (\$1 \ \$2 \ \$3)^T$ e o vetor de parâmetros desejados $p = (d \ \theta \ \alpha)^T$, e J_{qp} a matriz Jacobiana de transformação entre o vetor de coordenadas desejadas q e o vetor de parâmetros desejados p , tem-se que

$$\underbrace{\begin{pmatrix} \delta \$1 \\ \delta \$2 \\ \delta \$3 \end{pmatrix}}_{\delta \$} = J_{sp} \underbrace{\begin{pmatrix} \delta d \\ \delta \theta \\ \delta \alpha \end{pmatrix}}_{\delta p} \quad (75)$$

$$\underbrace{\begin{pmatrix} \delta x \\ \delta y \\ \delta \theta' \end{pmatrix}}_{\delta q} = J_{qp} \underbrace{\begin{pmatrix} \delta d \\ \delta \theta \\ \delta \alpha \end{pmatrix}}_{\delta p} \quad (76)$$

Com isso é possível formular a matriz que relaciona variações das coordenadas desejadas com variações do vetor de características do objeto

$$\begin{pmatrix} \delta x \\ \delta y \\ \delta \theta' \end{pmatrix} = J_{qp} J_{sp}^{-1} \begin{pmatrix} \delta \$1 \\ \delta \$2 \\ \delta \$3 \end{pmatrix} \quad (77)$$

sendo

$$J_{sp} = \begin{bmatrix} 1/K & 0 & 0 \\ 0 & \cos \theta & 0 \\ 0 & 0 & 1/r \end{bmatrix} \rightarrow J_{sp}^{-1} = \begin{bmatrix} K & 0 & 0 \\ 0 & \sec \theta & 0 \\ 0 & 0 & r \end{bmatrix} \quad (78)$$

$$J_{qp} = \begin{bmatrix} \text{sen } \theta & d \cos \theta & 0 \\ \cos \theta & -d \text{sen } \theta & 0 \\ \frac{1}{d} \tan(\theta - \theta') & 1 & -\frac{1}{d} \sec(\theta - \theta') \end{bmatrix} \quad (79)$$

Escrevendo as matrizes acima em relação aos novos parâmetros $\$1$, $\$2$ e $\$3$, tem-se que:

$$J_{sp}^{-1} = \begin{bmatrix} K & 0 & 0 \\ 0 & \frac{1}{\sqrt{1-\$2^2}} & 0 \\ 0 & 0 & r \end{bmatrix} \quad (80)$$

$$J_{qp} = \begin{bmatrix} \sqrt{1-\$2^2} & -K\$1\$2 & 0 \\ \$2 & K\$1\sqrt{1-\$2^2} & 0 \\ r\$3 & 1 & -\frac{1}{\sqrt{K^2\$1^2 - r^2\$3^2}} \\ \frac{r\$3}{K\$1\sqrt{K^2\$1^2 - r^2\$3^2}} & & \end{bmatrix} \quad (81)$$

Sendo assim, conhecidos os valores reais e desejados de $\$1$, $\$2$ e $\$3$, pode-

se determinar o vetor $\delta\$ = \begin{pmatrix} \$1 - \$1_d \\ \$2 - \$2_d \\ \$3 - \$3_d \end{pmatrix}$ e conseqüentemente, a partir da equação

(77), determinar os erros referentes às 3 juntas desejadas δx , δy , $\delta \theta'$. Note que a equação (77) só é exata para erros infinitesimais, portanto a sua aplicação a erros finitos é uma aproximação. Esta aproximação melhora à medida que a câmera se aproxima de seu objetivo quando os erros $\delta\$1$, $\delta\$2$ e $\delta\$3$ tendem para zero.

Vale ressaltar que neste caso a matriz Jacobiana de imagem consegue determinar o vetor de características independente das posições atuais dos motores, sendo assim o sistema não precisa de realimentação dos sensores de posição. Para um sistema genérico é necessário uma realimentação dos motores.

4.7. Equacionamento do sistema com alvo 2D genérico

Para a segunda parte do experimento, o sistema será modificado de forma a operar utilizando como referência alvos planos genéricos. Para fazer o reconhecimento das imagens obtidas pela câmera, será utilizado o algoritmo SIFT (descrito anteriormente na seção 2.2) responsável por determinar pontos chaves na imagem. Através deste algoritmo, também é possível criar pares de pontos correspondentes entre duas imagens de vistas diferentes de um mesmo objeto. A partir destas correspondências, poderão ser obtidas as distâncias relativas entre a

câmera e o objeto. Executando o algoritmo sobre uma imagem de referência, cujas coordenadas dos pontos no espaço são conhecidas, é possível determinar para cada ponto encontrado a relação entre as coordenadas deste na imagem (em *pixels*) e as coordenadas no espaço. Na Figura 33 é representado um objeto plano utilizado como alvo no espaço, onde os pontos indicados representam os pontos chaves encontrados pelo algoritmo. Um sistema de referência (x_o, y_o) é fixado ao objeto, com origem definida em seu centro geométrico.

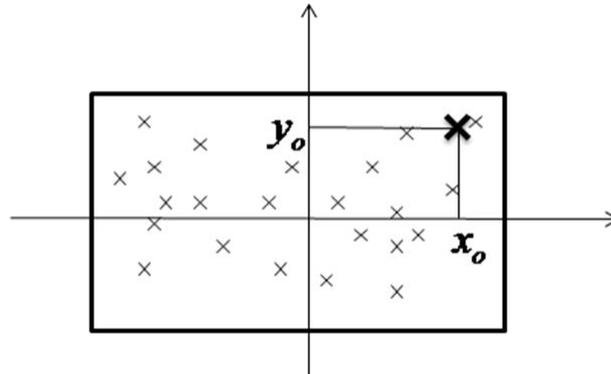


Figura 33 - Pontos chaves (x_o, y_o) encontrados pelo algoritmo SIFT sobre o objeto alvo

A Figura 34 mostra a projeção de um ponto do objeto, a uma distância x_o de seu centro na horizontal, considerando que a câmera se encontra a uma distância (x_c, y_c) do centro do objeto, e formando um ângulo θ entre seu eixo óptico e a direção de y_c .

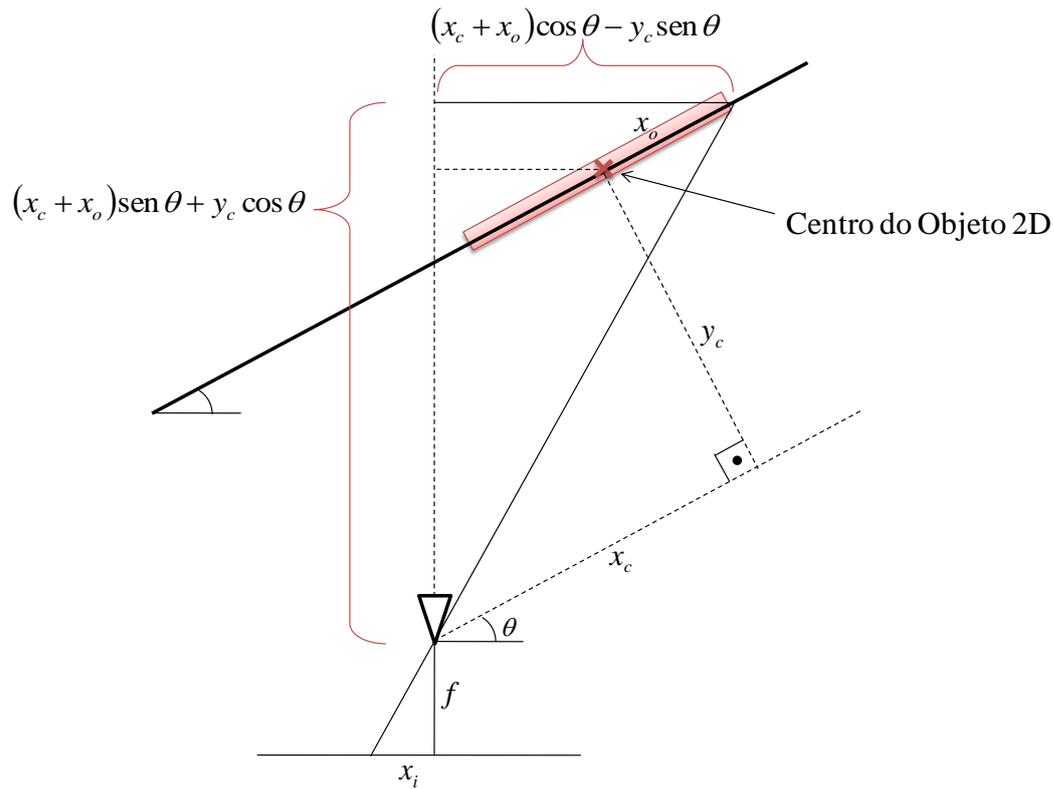


Figura 34 – Esquema do experimento utilizando objetos 2D

Na Figura 34, x_o representa a distância entre um dos pontos chaves e o centro do objeto alvo medida na horizontal. Já x_i representa a mesma distância nas coordenadas da imagem, em *pixels*. Sendo f a distância focal da câmera, expressa em *pixels*, tem-se que:

$$\frac{(x_c + x_o) \operatorname{sen} \theta + y_c \cos \theta}{f} = \frac{(x_c + x_o) \cos \theta - y_c \operatorname{sen} \theta}{x_i} \quad (82)$$

e conseqüentemente monta-se o sistema

$$[f \quad x_i \quad x_o \cdot x_i] \cdot \underbrace{\begin{pmatrix} y_c \tan \theta - x_c \\ x_c \tan \theta + y_c \\ \tan \theta \end{pmatrix}}_X = f \cdot x_o \quad (83)$$

Para M pares de pontos (x_o, x_i) , o vetor X pode ser determinado por um ajuste de mínimo quadrados

$$\underbrace{\begin{bmatrix} f & x_{i1} & x_{o1}x_{i1} \\ f & x_{i2} & x_{o2}x_{i2} \\ \vdots & \vdots & \vdots \\ f & x_{im} & x_{om}x_{im} \end{bmatrix}}_A \cdot X = f \cdot \underbrace{\begin{bmatrix} x_{o1} \\ x_{o2} \\ \vdots \\ x_{om} \end{bmatrix}}_B \Rightarrow A \cdot X = B \Rightarrow X = \text{pinv}(A) \cdot B \quad (84)$$

onde $\text{pinv}(A)$ é a pseudo-inversa da matriz A .

Uma vez determinado o vetor X , podem ser determinadas as distâncias relativas desejadas (x_c, y_c, θ) . Sendo $X = [X_1 \ X_2 \ X_3]^T$, obtém-se a partir da definição de X

$$\begin{aligned} \theta &= \text{atan}(X_3) + k\pi \\ x_c &= \frac{X_2 \tan \theta - X_1}{1 + \tan^2 \theta} \\ y_c &= \frac{X_1 \tan \theta - X_2}{1 + \tan^2 \theta} \end{aligned} \quad (85)$$

Note que as equações (84) e (85) dependem dos valores de x_{o1}, \dots, x_{om} para obter x_c, y_c e θ . Esses valores foram obtidos previamente a partir da imagem de referência, capturada em uma posição conhecida da câmera e aplicando a equação (82) para cada um dos M pontos SIFT da imagem de referência.

Os equacionamentos descritos acima foram realizados em função da coordenada x_o dos pontos do objeto. De forma análoga, as mesmas distâncias poderiam ter sido obtidas utilizando as coordenadas em y_o . A partir do sistema:

$$\begin{bmatrix} y_i & x_o \cdot y_i \end{bmatrix} \cdot \underbrace{\begin{pmatrix} x_c \sin \alpha + y_c \cos \alpha \\ \sin \alpha \end{pmatrix}}_{X^*} = F \cdot y_o \quad (86)$$

e utilizando a formulação da pseudo-inversa para M pares de pontos (y_o, y_i) :

$$\underbrace{\begin{bmatrix} y_{i1} & x_1 y_{i1} \\ y_{i2} & x_2 y_{i2} \\ \vdots & \vdots \\ y_{im} & x_m y_{im} \end{bmatrix}}_{A^*} \cdot X^* = f \cdot \underbrace{\begin{bmatrix} y_{o1} \\ y_{o2} \\ \vdots \\ y_{om} \end{bmatrix}}_{B^*} \Rightarrow X^* = \text{pinv}(A^*) \cdot B^* \quad (87)$$

o vetor X^* pode ser determinado. A partir de $X^* = [X_1^* \ X_2^*]^T$, é possível apenas determinar o par de equações

$$\begin{aligned}\alpha &= \text{asen}(\mathbf{X}_2^*) + k\pi \\ x_c \sin \alpha + y_c \cos \alpha &= \mathbf{X}_1^*\end{aligned}\quad (88)$$

Note que ao usar apenas (y_o, y_i) para estimar a pose da câmera em relação ao objeto, obtém-se um sistema indeterminado, com apenas 2 equações para as 3 incógnitas x_c , y_c e θ . Isto ocorre porque o eixo óptico da câmera está sempre em um plano horizontal perpendicular à direção vertical y_o . Assim, na prática apenas os pares (x_o, x_i) são usados na obtenção de (x_c, y_c, θ) . As equações (86-88) são usadas apenas para conferir os resultados.

No capítulo seguinte, serão descritos os testes realizados tanto com o alvo em forma de círculo quanto com a imagem do painel, e serão apresentados gráficos, quantificando os resultados. Por fim, será apresentada uma comparação entre as técnicas *look-and-move* e servo-visual mostrando as vantagens e desvantagens de cada uma.